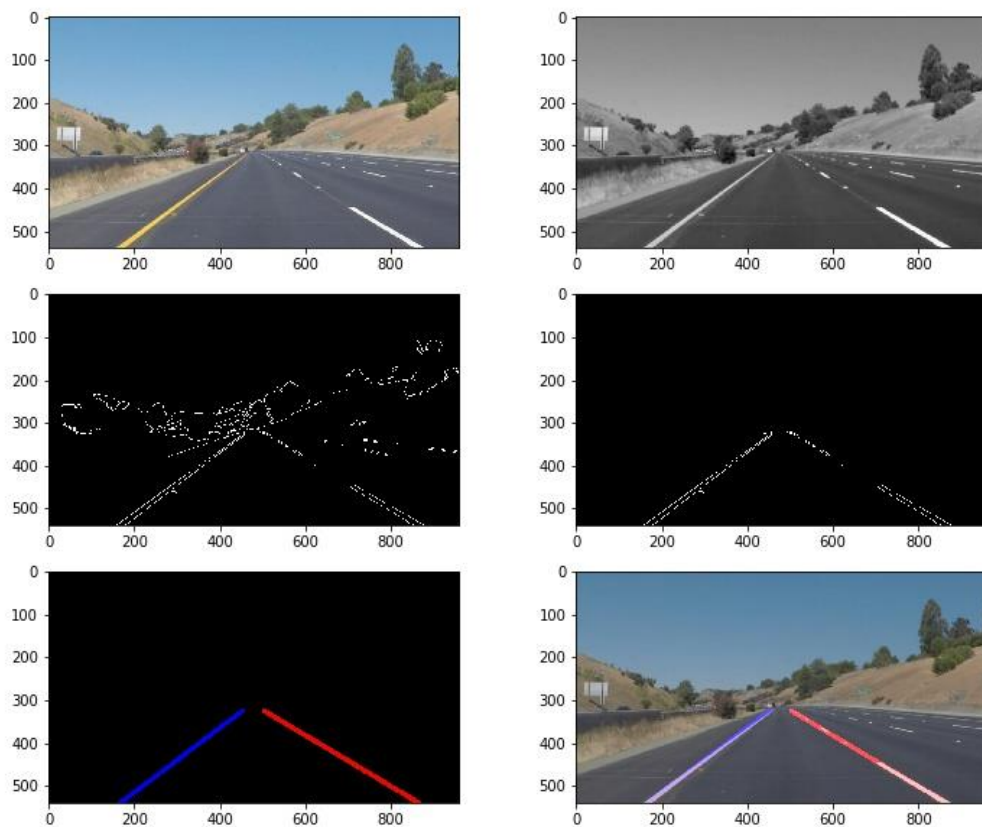


# Self-Driving Car Engineer Nanodegree

## Project: Finding Lane Lines on the Road

The goal of this project is designing an image processing pipeline, that will identify the driving lanes in images / video frame and will annotate the left and right lane.



## Methodology

My image processing pipeline consist of the following 6 steps.

1. Convert image to grey scale
2. Apply gaussian blur to smoothen the gradients.
3. Apply canny edge detection algorithm to detect the edges.

4. Identify and apply the polygon mask to filter the edges to eliminate noise.
5. Apply Hough transform to identify the line segments in the filtered image.
6. Separate the left and right lanes, and apply first degree polynomial fit to the points to extend the lanes.

## Procedure for left and Right Lanes

In order to separate the two lanes, I modified the draw line helper function. For each line identified by the Hough line, I calculated the slope and segregated them to left lane and right lane basis the following thresholds.

**slope > 0.2: right lane**

**slope <=0.2: left lane**

A new function **draw\_line** was created to delegate the extrapolation of the hough lines. We used the np.polyd function to generate the x values.

## Pipeline Shortcomings

1. The pipeline is not stable while the image has curvature in lanes in both video and image.
2. The line size is not always consistent.
3. The algorithm is some time capturing the noise.

## Pipeline Improvements

1. Improve the line fit so that it accounts to road curvature, may be using higher degree polynomial fit.
2. Implement robust image filtering by utilizing HSV color space